

Unsupervised Spatio-Temporal Embeddings for User and Location Modelling

Jing Yang
ETH Zurich
Rämistrasse 101
Zurich, Switzerland 8092
jing.yang@inf.ethz.ch

Carsten Eickhoff
ETH Zurich
Rämistrasse 101
Zurich, Switzerland 8092
carsten.eickhoff@inf.ethz.ch

ABSTRACT

Many social network applications depend on robust representations of spatio-temporal data. In this work, we present an embedding model based on feed-forward neural networks which transforms social media check-ins into dense feature vectors encoding geographic, temporal, and functional aspects for modelling places, neighborhoods, and users. On the basis of this model, we further propose a Spatio-Temporal Embedding Similarity algorithm (STES) for location recommendation.

In a range of experiments on real life data collected from Foursquare, Twitter, and Gowalla, we demonstrate our model's effectiveness at characterizing places and people, resulting in a significantly increased recommendation and classification performance as compared to the state of the art. Finally, we select eight major cities around the globe and verify the robustness and generality of our model by porting pre-trained models from one city to another, and thereby alleviating the need for costly local training.

CCS CONCEPTS

•**Networks** → Location based services; •**Information systems** → Recommender systems;

KEYWORDS

Social networks, Check-in embedding, Personalized location recommendation, Crime prediction

ACM Reference format:

Jing Yang and Carsten Eickhoff. xxxx. Unsupervised Spatio-Temporal Embeddings for User and Location Modelling. In *Proceedings of xxx, xxx*, xxx, 11 pages. DOI: xxx.xxx

1 INTRODUCTION

Spatial social network applications and services, such as transport network design, location-based services, and crime prediction, normally involve two important issues: understanding residents' real-time activities and describing urban spaces accurately [5, 7, 33]. Towards the former item, people usually utilize check-in data from social network platforms (e.g., Twitter, Foursquare, or Facebook),

which cover a significant number of users in the form of check-ins and comments about points-of-interest (POIs). For the latter aspect, place annotation approaches are required. A common method is to simply annotate places with categorical labels such as *Home*, *Restaurant*, and *Shop* [8, 14, 32, 43]. However, it is unclear whether such discrete tags offer sufficient flexibility and descriptive power for modelling the complexity of urban landscapes.

In this work, we represent locations by means of embedding vectors in a semantic space. Our framework solely relies on social media check-ins and the embedding model is built conceptually based on the *Word2Vec* technique [26]. Aiming to annotate locations in terms of temporal, geographic, and functional aspects, we decompose the time, the locations, and the venue functions of check-in records into virtual "words". We then treat check-in sequences as "sentences" and regard activity profiles of a neighborhood or a user as "documents".

In comparison with the traditional discrete method, our approach describes places in a continuous manner and preserves more information about people's real behavior as well as places' day-to-day usage patterns. For instance, in the case of label annotation, three food related places which serve Chinese breakfast, pizza, and sushi respectively may all be labeled as *Restaurant* but their features in food type, active hours, and location may vary dramatically. In the course of this paper, we will see how our embedding model represents such within-class variance in a natural way. As the embedding vectors are learnt from people's real-time check-ins, we also leverage them in user representation to reflect people's activity patterns and interests.

The embedding model is an accurate descriptor of places and users in terms of geographic and functional affinity, activity preference, and daily schedules. As a consequence, it can be applied in a wide range of settings. In this paper, we elaborate two practical applications: *location recommendation* and *crime prediction*. The novel contributions of this work are threefold:

- We present an **unsupervised** spatio-temporal embedding scheme based on social media check-ins. Trained with monthly check-in sequences, the model shows wide applicability for tasks ranging from social science problems to personalized recommendations.
- Based on this model, we propose the **STES** algorithm that recommends locations to users. Compared with state-of-the-art recommendation frameworks, we can achieve an improvement up to 30%.
- The model shows strong **robustness** and **generality**. Once trained in a representative city, it can be directly utilized in other cities with only slight generalization errors (< 3%).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

xxx, xxx

© xxxx Copyright held by the owner/author(s). xxx.xxx...\$15.00
DOI: xxx.xxx

The remainder of this paper is structured as follows: Section 2 gives an overview of related work on representation learning of spatio-temporal data for location recommendation and crime prediction. Section 3 describes our embedding model and the STES location recommendation algorithm. Section 4 empirically evaluates the performance of our model on a number of tasks and data sets comparing to a competitive range of performance baselines. Finally, Section 5 concludes with a summary of our findings and a discussion of future work.

2 RELATED WORK

In this section, we first review *Word2Vec* and its applications on basis of social networks. Then, we introduce the state-of-the-art in our two application domains, location recommendation and crime prediction.

Word2Vec and Applications. Intended for natural language processing, *Word2Vec* [26] processes textual documents in an unsupervised manner and learns embedding vectors for words and phrases based on their context of occurrence in text. Due to its efficiency in capturing the contextual correlation of items, researchers have been applying the technique for E-commerce product recommendation [30], network embedding [29], user profiling [35, 36], social media prediction tasks [39, 42], and many others. When employed in social network context, however, the model is still mainly used on texts such as tweets and Yelp reviews. Our analogy is to treat check-in sequences as virtual “sentences” and users or neighborhoods as “documents”. Consequently, correlations of contextual locations and activities can be better modelled.

Location Recommendation. As the most common performance measure for spatial models [2, 44], location recommendation has been popular in recent years in studies on location-based social networks (LBSNs). The most basic location recommendation approaches are content based, relying solely on properties of users and locations [22, 37]. The recurring idea is to explore user and location features, then make recommendations based on their similarities and past preferences. Another popular branch of approaches employs matrix factorization (MF) [13, 31] and its derivative methods [2, 3, 18, 19]. MF-related methods aim to represent users and items in matrices and to recommend locations based on the row-to-row correlation. Methods based on topic models (TM) [1] and Markov model (MM) [23] also perform well in recommendation tasks. Often, geographic and temporal information are included as additional evidence [6, 9, 10, 15, 17, 45].

Recently, some recommendation methods [21, 28, 46] involve *Word2Vec* in their processing schemes. However, they only focus on geographic location modelling but ignore other information (e.g., venue function, check-in time) associated with each check-in. Consequently, these models hold very limited applicability beyond the immediate context of location recommendation that they are designed for. There also exist some recommendation approaches involving LSTM recurrent neural networks [16, 34]. However, such methods are into the class of supervised learning. In contrast, we develop an unsupervised learning approach including manifold check-in information which can be utilized in a wide range of settings by giving a robust representation of each location and does not require supervised label information. Section 4 will highlight

Table 1: Time Discretization

Timeslot Tag	Time
Morning/WeekendMorning	6:00AM-10:59AM
Noon/WeekendNoon	11:00AM-1:59PM
Afternoon/WeekendAfternoon	2:00PM-5:59PM
Evening/WeekendEvening	6:00PM-9:59PM
Night/WeekendNight	10:00PM-5:59AM

the resulting performance difference both in the context of recommendation as well as other tasks.

Crime Prediction. Crime prediction is a typical social science issue. Iqbal *et al.* [11] rely on demographic features such as population, household income, and education level as input and predict regional crime rates on a three-point scale. Gerber [7] and Wang *et al.* [40] design topic models based on Twitter posts to predict criminal incidences. In this work, we demonstrate that people’s day-to-day activity patterns are strong indicators of crime occurrence. Based on the same embedding model as described above, we characterize neighborhoods with check-in vectors. Acting as features, these vectors perform well in future crime rate and crime occurrence prediction.

3 METHODOLOGY

In this section, We begin by introducing the problem domain. Then, we describe the check-in embedding model. Finally, we propose the location recommendation algorithm STES.

3.1 Scenario

As the only input data, social media check-ins are first grouped into “words”, “sentences”, and “documents” for embedding training. Important definitions are proposed as follows.

Check-ins. A check-in is a tuple $c = \langle u, f, t, l \rangle$ which depicts that a user u visits a location l at time t and f demonstrates the functional role of the visited venue. We discretize t in 10 timeslots, namely, *Morning*, *Noon*, *Afternoon*, *Evening*, *Night*, *WeekendMorning*, *WeekendNoon*, *WeekendAfternoon*, *WeekendEvening*, and *WeekendNight*. Discretization thresholds are listed in Table 1. This definition is motivated by a natural reflection of daily routines. *Morning* is the time bucket when people start a day and work before lunch. *Noon* is the lunch break time. *Afternoon* refers to working hours after lunch. *Evening* is for dinner and after-work activities. *Night* corresponds to the sleeping hours. As daily activities on weekends are usually different from those on weekdays (e.g., *Afternoon* is often for leisure instead of work), we additionally define a set of timestamps on weekends that has the same hour correspondence. A check-in’s functional role f is described following the popular Foursquare venue categories¹.

Considering the check-in meaning, check-in size, and data set scale, for each check-in record, we concatenate functional role f and timeslot t as its *feature word* (e.g., “*Bar-Evening*”), and use the unique id of location l as its *geographic word* (e.g., “423e0e80f964a52-044201fe3”).

Check-In Sequences. A check-in sequence contains a *feature word* sequence and a *geographic word* sequence. Words in these

¹<https://developer.foursquare.com/categorytree>

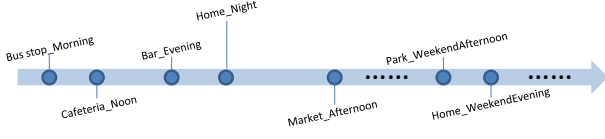
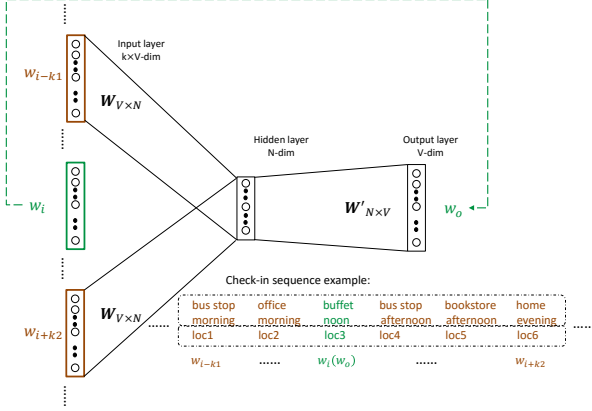
Figure 1: A check-in sequence of *feature words*.

Figure 2: Embedding training framework. Each word takes turn to be predicted from context words.

two sequences are one-to-one correspondent. Each sequence is a chronological ordering of check-ins in one month of the profile of a user u or a neighborhood n . Figure 1 shows a *feature word* sequence example.

Users/Neighborhoods. Dependent on whether the task is user centric (e.g., location recommendation) or area centric (e.g., crime prediction), all check-ins of a user u or in a neighborhood n are regarded as a document.

Our model learns embedding vectors of *feature words* and *geographic words* given check-in sequences. Then we can calculate representations for locations, neighborhoods, and users.

3.2 Embedding Model

Figure 2 illustrates the neural network (NN) embedding training framework. Each target word $w_i(w_o)$ is predicted from input context words which are processed in a sliding window from w_{i-k1} to w_{i+k2} . $k1$ and $k2$ are adjustable and $k = k1 + k2$ is the context window size. The words are initialized with one-hot encoded vectors, which means for a given word, only one out of V vector components will be 1, and all the others are 0. When the training is finished, the row of weight matrix $W_{V \times N}$ from input layer to hidden layer is the vector representation of the corresponding word.

The training objective is to minimize the loss function

$$E = -\log p(w_o | w_{i-k1}, \dots, w_{i+k2}) \quad (1)$$

$p(w_o | w_{i-k1}, \dots, w_{i+k2})$ is the probability of the target word given those context words, which can be formulated as a softmax function

$$p(w_o | w_{ik}) = \frac{\exp(\mathbf{v}'_{w_o} \mathbf{v}_{w_{ik}})}{\sum_{j=1}^V \exp(\mathbf{v}'_{w_j} \mathbf{v}_{w_{ik}})} \quad (2)$$

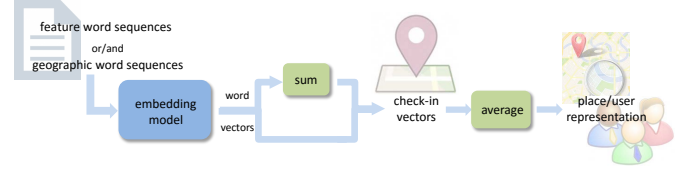


Figure 3: Flowchart from importing check-ins to obtaining embedding vectors of places or users.

where,

$$\mathbf{v}_{w_{ik}} = \frac{1}{k} (\mathbf{v}_{w_{i-k1}} + \dots + \mathbf{v}_{w_{i+k2}}) \quad (3)$$

In Equation (2), \mathbf{v}'_w s come from columns of $W'_{N \times V}$, which is the weight matrix from hidden layer to output layer. Back-propagation is applied during the training process and both hidden-to-output weights (W') and input-to-hidden weights (W) are updated using stochastic gradient descent.

Also from Equation (2), we can see that the learning process involves a traversal of the whole vocabulary, which greatly jeopardizes model efficiency. To tackle the problem, we employ the hierarchical softmax algorithm as proposed in [25]. To do so, we build a Huffman binary tree [12], in which V vocabulary words are leaf units, and for each of them, there exists a unique path to the root. We only consider the along-path words when calculating the loss function.

Feature words and *geographic words* are separately trained via this model, resulting in a feature embedding space and a geographic embedding space. Now we can represent a check-in c by only its *feature word* vector or only its *geographic word* vector or completely by summing up its *feature word* vector and *geographic word* vector.

Furthermore, a user u can be represented by the mean of his/her check-in vectors, which also works if we only want to profile their activity in a specific time window. The same approach is applied to annotate a location or a neighborhood. Figure 3 demonstrates the entire workflow.

To conclude the *word2vec* analogy that motivates our network training process, we can conceptually think of *feature words* and *geographic words* of check-in records as “words”, check-in sequences as “sentences”, and check-in profiles of users or neighborhoods as “documents”.

3.3 Recommendation Algorithm

Our recommendation algorithm is based on the user-location cosine similarity in the newly established embedding space. Recall that in the literature review, we mentioned how both temporal and geographic elements play important roles in recommendation tasks. Therefore, we utilize both *feature word* and *geographic word* vectors to make recommendations. In this case, a check-in is represented (\mathbf{v}_c) by the element-wise summation of its two vector representations ($\mathbf{v}_{f_w}, \mathbf{v}_{g_w}$)

$$\mathbf{v}_c = \mathbf{v}_{f_w} + \mathbf{v}_{g_w} \quad (4)$$

Based on these vector representations, we profile locations and users. Remember that functional roles are defined by social network

venue categories. For the sake of more straightforward understanding, we will refer to “venue category” in place of “functional role” in the rest of this section.

Location Profile. Although two locations may be of the same venue category, they can still be differentiated if they are usually visited in different timeslots. A location l can thus be represented as (\mathbf{v}_l) by averaging all user check-ins (\mathbf{v}_c) issued there:

$$\mathbf{v}_l = \frac{1}{M} \sum_{m=1}^M \mathbf{v}_{c_m} \quad (5)$$

where M is the total count of check-ins originating from location l .

User Profile. In different timeslots, users show different check-in preferences in terms of venue categories and geographic areas. Therefore, in each timeslot t , we represent a user u (\mathbf{v}_{u_t}) by averaging his/her check-ins (\mathbf{v}_{c_t}) and calculating a user coordinate centroid $(coordinate_{u_t})$ from his/her check-in locations $(coordinate_{c_t})$.

$$\mathbf{v}_{u_t} = \frac{1}{N} \sum_{n=1}^N \mathbf{v}_{c_{t,n}} \quad (6)$$

$$coordinate_{u_t} = \frac{1}{N} \sum_{n=1}^N coordinate_{c_{t,n}} \quad (7)$$

where N is the total count of check-ins from the user u in a timeslot t .

Next, we calculate two *cosine similarities* for users in each timeslot. The first one, *user-activity similarity* (S_{u-a}), is between the user vector (\mathbf{v}_{u_t}) and all check-in vectors from the whole vocabulary of the user $(\mathbf{v}_{c,u})$, which indicates the user-preferred venue categories. The second one, *user-location similarity* (S_{u-l}), is between the user vector (\mathbf{v}_{u_t}) and all location vectors (\mathbf{v}_l) , which indicates the user-preferred locations in each venue category.

During the recommendation stage, given a timeslot t , cosine similarities S_{u-a} and S_{u-l} , we first select the C most favored venue categories of the user in descending order f_1, \dots, f_C , then we focus on the locations in selected categories. For each location, we calculate its distance (*dist*) to the user coordinate centroid in this timeslot ($coordinate_{u_t}$). Considering the category preference order and location-to-centroid distances, we introduce two exponential decay factors, *category decay* (CD) and *spatial decay* (SD).

$$CD = a_1 \times \exp(-a_2 \times f_c) \quad (8)$$

$$SD = b_1 \times \exp(-b_2 \times dist) \quad (9)$$

Where, $a_1, a_2, b_1, b_2 \in \mathbb{R}$, $f_c \in \{0, 1, \dots, C-1\}$. Therefore, final user-location similarity is calculated as

$$S_{u-l,final} = S_{u-l,original} \times CD \times SD \quad (10)$$

Then, we sort all locations belonging to these C categories in descending order of $S_{u-l,final}$ and make recommendations from the top. We refer to this algorithm as *Spatial-Temporal Embedding Similarity algorithm* (STES) and use the acronym STES in the rest of the paper.

4 EXPERIMENTS & EVALUATION

In this section, we begin by introducing the experimental data sets and data pre-processing details, then we elaborate on the various

experiments and evaluate the results. Our empirical investigation is driven by the following research questions:

RQ1. How well does the embedding model differentiate locations and users along temporal, geographic, and functional aspects?

RQ2. How does the STES algorithm perform in location recommendation compared with state-of-the-art recommendation methods?

RQ3. How well can we predict typical urban characteristics based on the embedding model?

RQ4. With what generalization error can an embedding model trained in one city be transferred to other cities?

4.1 Data set

We rely on a publicly available data set from [4] which contains check-ins from Foursquare, Twitter, Gowalla, and other location based social networks. This data set is global and covers 11 months from Mar. 2010 to Jan. 2011. Most of the check-ins are distributed in the U.S. and we choose New York City (NYC) as the representative city for the first part of our study.

To define functional roles of venues, we use the second level Foursquare venue categories, containing 422 classes such as *American Restaurant*, *Bar*, and *Metro Station*. This level achieves the best sparsity-specificity trade-off in comparison with the top level which is too coarse and the third or the fourth levels which are too specific to cover all venues.

We find that sometimes users repeatedly check in at the same venue in an artificially short time period although they remain in an unchanged activity. This is reasonable as check-ins are often posted in a casual way in which people share real-time affairs and moods. This phenomenon is especially common in recreational and eating activities. For instance, a user may check in for several times during one meal with friends, each time posting a newly served dish or comments about the food. To model activity sequences most reliably, we delete repeated check-ins from an individual and retain only the first check-in at this location. To reduce noise, we further remove both users and locations with less than 10 posts. In the end, our NYC data set contains 225,782 check-ins by 6,442 users at 7,453 locations.

To define urban “neighborhoods” for area centric task *crime prediction*, we utilize the Census Block Group (CBG) polygons² defined in the year of 2010, which matches the time period of the check-in data. A CBG may contain several Census Blocks (CBs), which are the smallest geographic areas that the U.S. Census Bureau uses to collect and tabulate census data. With boundaries defined by physical streets, streams, and railroad tracks as well as invisible town limits, property lines, and imaginary extensions of streets, these polygons represent the most natural segmentation of a city. There are 6,493 CBGs in NYC and 1,720 of them are populated by the denoised check-ins.

4.2 Qualitative Analysis of Embedding Vectors

The embedding vectors were designed to retain activity and location features so that users and places are differentiable. During the training process, we set the embedding dimension to be 200. Figure 4 shows a downscaled visualization in 2-dimensional space.

²<http://data.beta.nyc/dataset/2010-census-block-groups-polygons>

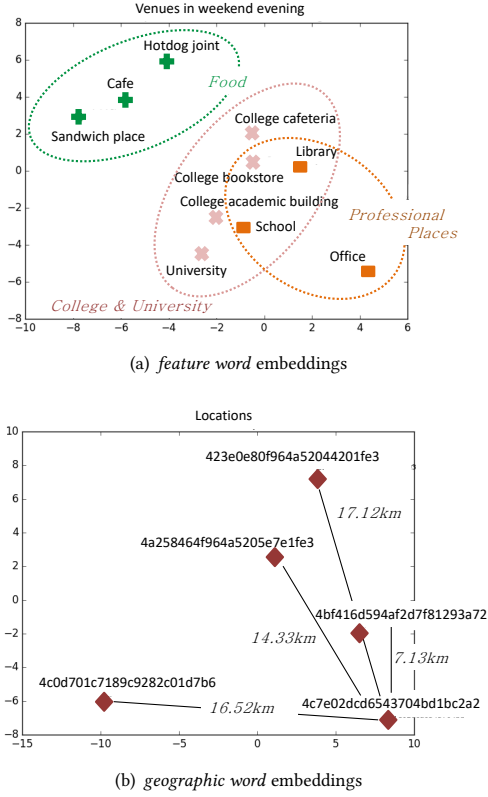


Figure 4: Feature word and geographic word example vectors in their semantic spaces.

As expected, *feature word* vectors and *geographic word* vectors are embedded with functional, temporal, and geographic similarities. Figure 4(a) shows some venue embeddings during weekend evenings. *Food* venues, *College&University* venues, and *Professional* venues are clustered into three groups with the latter two sharing a considerable overlap. This demonstrates the embedding’s ability to retain functional correlations that might not have been expressed by the raw venue category labels. More specifically, *School* and *Library* are much closer to *College&University* places than other professional places such as *Offices* that do not hold an educational intent.

Geographic word embeddings reflect geographic proximity among venues. In Figure 4(b), we take location *4c7e02dcd6543704bd1bc2a2* as an example. It can be seen that vector-to-vector similarities qualitatively correspond to location-to-location distances.

Such embedding vectors help to answer **RQ1**. Places and users are annotated in a way that reflects location visiting patterns as well as people’s activity preferences. In the following, we will quantitatively show how the embedding vectors can be used for specific problems.

4.3 Location Recommendation

The goal of location recommendation is to predict a top-k list of locations that a specific user might visit given a reference timeslot.

For each user, we choose his/her first 80% check-ins as training data and the remaining 20% as test data. For comparison, we include five state-of-the-art location recommendation algorithms as performance baselines.

STT [10]: This algorithm is based on spatio-temporal topic models. It learns the spatio-temporal parameters of users, topics, regions, and locations for recommendation.

GT-SEER [46]: This algorithm is based on *Word2Vec*. Considering geographic and temporal factors during the training process, it trains location id embeddings and models user preference through pairwise ranking.

TA-PLR [21]: This is a time-aware personalized location recommendation approach. Also based on *Word2Vec*, it first trains embeddings for location ids only, and further incorporates temporal factors in the recommendation algorithm.

Rank-GeoFM [17]: This is a ranking based factorization method, which includes geographic and temporal influence in a latent model.

LRT [6]: This model studies temporal influence and uses matrix factorization to capture user-location preference in different timeslots.

We compare the performances through *precision*, *recall*, *accuracy*, and *mean average precision (MAP)* as they are generally used in location recommendation systems [10, 20, 46]. We denote these metrics at top-k recommendation as $pre@k$, $rec@k$, $acc@k$, and $MAP@k$ respectively. Their definitions are formulated as follows,

$$pre@k = \frac{1}{|U|} \sum_{u \in U} \frac{|L_{gt} \cap L_{rec}|}{k} \quad (11)$$

$$rec@k = \frac{1}{|U|} \sum_{u \in U} \frac{|L_{gt} \cap L_{rec}|}{|L_{gt}|} \quad (12)$$

$$acc@k = \frac{|L_{gt,all} \cap L_{rec,all}|}{|T|} \quad (13)$$

$$MAP@k = \frac{\sum_{t \in T} AveP(t)}{|T|} \quad (14)$$

in which, U represents the user set, L_{gt} and L_{rec} represent the set of ground truth locations and the set of corresponding recommended locations for one user in the test data. $L_{gt,all}$ and $L_{rec,all}$ follow the same meanings but for all the users. T represents the total test data set, and $AveP(t)$ refers to the average precision for each test case.

We report the performances over all of the recommendation methods in Table 2. Table 2 lists the best performances of all the methods corresponding to the following parameters:

STT: Counts of latent regions and topics are both 150.

GT-SEER: Distance threshold is $1km$; 50 unchecked locations, β/α is 0.5; embedding size is 200.

TA-PLR: α is 1; λ is 0.01; embedding size is 200.

Rank-GeoFM: $\alpha, \gamma, \epsilon, C, K$ are 0.15, 0.0001, 0.3, 1, 100; 300 nearest neighbors.

LRT: Latent d is 10; α, β, λ are 2, 2, 1.

STES: Embedding size is 200; 15 most preferred venue categories; decay parameters a_1, a_2, b_1, b_2 are 0.4, 0.025, 1, 0.95.

With respect to **RQ2**, we find that our STES algorithm outperforms the baselines with respect to all the performance metrics. The advantage is especially significant in top-1 recommendation.

Table 2: Location Recommendation Performance evaluated by precision, recall, accuracy, and MAP.

	pre@1	pre@5	pre@10	rec@1	rec@5	rec@10	acc@1	acc@5	acc@10	MAP@1	MAP@5	MAP@10
LRT	0.371	0.105	0.054	0.016	0.038	0.052	0.017	0.033	0.059	0.017	0.029	0.042
Rank-GeoFM	0.42	0.152	0.069	0.019	0.046	0.067	0.021	0.051	0.077	0.021	0.045	0.058
GT-SEER	0.462	0.179	0.099	0.021	0.058	0.076	0.047	0.088	0.126	0.047	0.071	0.093
TA-PLR	0.457	0.184	0.096	0.025	0.062	0.087	0.045	0.101	0.143	0.045	0.074	0.091
STT	0.547	0.209	0.125	0.032	0.071	0.09	0.055	0.137	0.174	0.055	0.084	0.098
STES	0.606	0.227	0.147	0.064	0.089	0.109	0.105	0.176	0.199	0.105	0.128	0.132

We confirm the statistical significance of performance differences between our method and all of the contesting baselines using McNemar’s test [24]. The largest mid-p-value is 1.53×10^{-4} .

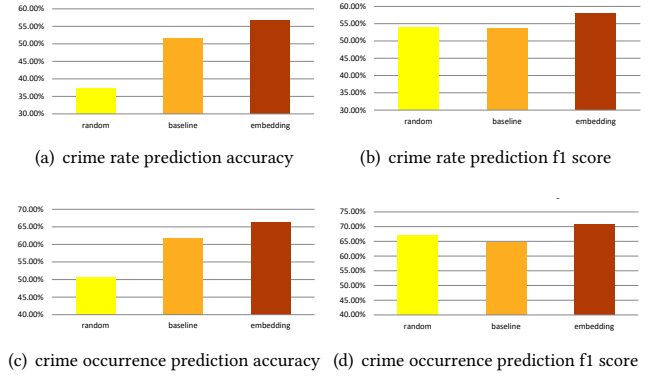
The results demonstrate the effectiveness of the embedding model and the STES algorithm in user/place characterization for recommending locations. As geographic and temporal aspects are all considered in these six approaches in different stages, we argue that our improvement mainly comes from the embedding of venues’ functional roles. In addition to indicating *where* and *when* someone is based on geographic and temporal messages, the functional information further explains *why* someone is there at that time, and this essentially reveals a person’s activity preference in addition to specific location preference. Consequently, we have better modelling power when a user is in a region which is away from his/her frequently visiting area. Moreover, calculating the *mean* of check-in vectors further leverages the continuity and smoothness of the embedding model and thus carries more latent correlations between users and locations.

4.4 Crime Prediction

By providing spatio-temporal embeddings for user and location characterization, our model represents a proxy for the social interactions observed in an urban area. In this section, we will further investigate this capability by addressing a well-known social science problem: crime prediction. Previous work [38, 41] demonstrated that the occurrence of criminal activities are correlated with place types and time, which are both encoded in our check-in embeddings.

For crime prediction, neighborhoods rather than users are our study subjects, and we only utilize *feature word* embeddings to characterize neighborhoods. This modification is motivated by the fact that *geographic words* are only locally descriptive. Therefore, a neighborhood cannot be described without prior training data from the exact location, but such new neighborhood prediction is possible if only modelled with the universally applicable *feature word* vectors, as long as visited venues have the same functional roles. NYC is still the representative city for study and the crime data originates from the NYC Open Data portal³.

4.4.1 Crime Rate Prediction. Let us begin by defining the task of future crime rate prediction [7, 40] as predicting the next-month crime rate of a neighborhood. We characterize each neighborhood monthly using the mean of check-in vectors in that month, and we assign crime incidents into neighborhoods according to their location coordinates. A neighborhood is labeled as “Low” crime

**Figure 5: Average crime prediction accuracies and f1 scores in NYC.**

rate (occurrences=0/month/neighborhood), “Medium” crime rate ($0 < \text{occurrences} < 3/\text{month/neighborhood}$), or “High” crime rate ($\text{occurrences} \geq 3/\text{month/neighborhood}$). Averaging over neighborhoods in each month, 31.3% are of “Low”, 37.2% are of “Medium”, and 31.5% are of “High” and the variance is 0.047 across months. Averaging across months per location, a neighborhood has a 34.1% chance of “Low”, a 39.3% chance of “Medium”, and a 26.6% chance of “High” with a variance of 0.087.

We take data from Mar. 2010 to Oct. 2010 (9983 neighborhoods) as a training set while those in Nov. and Dec. 2010 (2151 neighborhoods) represent our test set. To define a performance baseline, we use l2-normalized monthly counts of *feature words* as the baseline features since previous works [27, 47] succeeded in characterizing neighborhoods by normalized counts of containing venue categories and visiting timeslots. Although we also reviewed several prediction schemes in the literature part, they are not feasible in our case due to the lack of tweet texts and demographic statistics like *education level*.

We evaluate the performance through accuracy and f1-score and the results are shown in Figure 55(a) and Figure 55(b). After testing various classification frameworks, we use a random forest classifier. The leftmost bar “random” refers to the results from consistently predicting the most likely label based on the ground truth of the training data. In this case it is “Medium”. The figures show that our embedding model produces the best results according to both metrics.

4.4.2 Crime Occurrence Prediction. Aside from predicting overall crime rates, we are interested in understanding local crime

³<https://data.cityofnewyork.us/Public-Safety/NYPD-7-Major-Felony-Incidents/hyij-8hr7>

hazards in greater detail. In our crime data set, a frequently occurring crime type in NYC is *Grand Larceny*. We will now investigate whether we can predict the occurrence of this particular type of crime in a neighborhood within next month. Experimental settings and classifier choice remain the same while the labels are changed into “No Grand Larceny” and “Grand Larceny”. When averaging over neighborhoods in each month, the label ratio is 51% for “No Grand Larceny” and 49% for “Grand Larceny” with a variance of 0.027. For each neighborhood, it has on average a 60.9% probability that this crime would occur with a variance of 0.082.

Figure 55(c) and Figure 55(d) demonstrate the average prediction accuracies and f1-scores. Similar to crime rate prediction, we can observe that our model outperforms both random guessing and the baseline.

Both crime rate and occurrence prediction results pass the McNemar’s significance test with the largest mid-p-value of 1.2062×10^{-7} . To justify the rationale behind our method and results, we further examine the check-ins in neighborhoods with low/medium/high crime rate and with/without grand larceny, and we plot their check-in time and location category distribution in Figure 6. We can see that neighborhoods with high crime rate are more checked in on weekdays at entertainment (e.g. casino), shopping, and professional (e.g. business center) places, which also applies to neighborhoods with more grand larceny cases. In response to **RQ3**, this section demonstrates the latent correlation between people’s daily activities and crime occurrence in an urban area. The prediction results further support our embedding model’s effectiveness in characterizing places.

4.5 Model Generalization

The training of large-scale embedding models can be a costly process requiring hours of compute time. To save this time, we investigate whether a pre-trained model can be directly applied in other cities while maintaining its performance. We use NYC as the reference city where the embedding model is trained, and select eight major global cities from the same data set for generalization test. After pre-processing, eight cities and their check-in statistics are listed in Table 3. We also list NYC statistics for comparison.

4.5.1 Generalization of Location Recommendation. Recall that in our earlier investigation in Section 4.3, we rely on both *feature words* and *geographic words* for location recommendation. However, since *geographic words* are locally descriptive, they cannot be easily taken out of their original frame of reference. As a consequence, we only generalize the NYC-based embedding model for *feature words*, but locally train *geographic words*.

Upon careful examination, none of the baseline methods can be ported across cities as they exclusively focus on the local location id embeddings (**GT-SEER** and **TA-PLR**), vectorization (**Rank-GeoFM** and **LRT**), or modelling (**STT**). Therefore, we have to train the baseline methods locally when applying them to different cities.

As before, we measure the performances through *precision*, *recall*, *accuracy*, and *MAP* and the results are listed in Table 5. Except for Seattle, local STES models consistently perform the best in all the cases in the other seven cities, followed by NYC-based STES and the baseline methods; while in Seattle, NYC-based STES model produces very close results compared with the local STES model

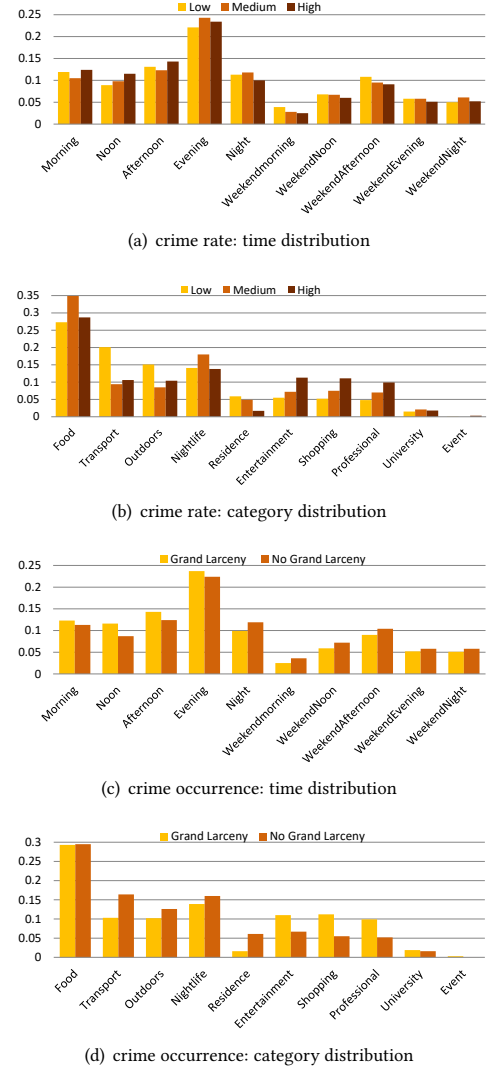


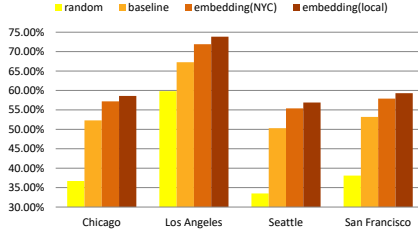
Figure 6: Check-in time and location category distribution in neighborhoods of different labels.

and the former outperforms the latter according to *recall* at top-1 recommendation.

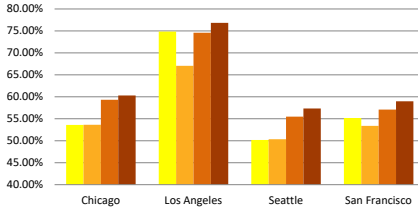
Upon closer examination, in all of the eight cities, the gaps between STES(NYC) and STES(local) are generally lower than 3%. In particular, these two methods almost tie at top-1 recommendation for U.S. cities; while in the other four non-US cities, NYC-based STES model produces less satisfactory performances with respect to their local models. We argue that the geographic distance from NYC reflects the cultural difference and this is the main reason for the performance differences. Specifically, life style in Southeast Asia is distinct from that in the U.S., which also applies for Europe while the dissimilarity is smaller. Therefore, the NYC-based embedding model is well adapted to other U.S. cities but somewhat less competent in European and Asian cities.

Table 3: Check-In Statistics

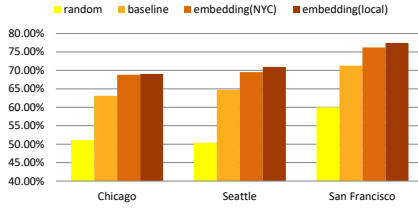
City	# of check-ins	# of users	# of locations	ave. check-ins/user (density)
Chicago (CH)	86,117	2,755	3,678	31.26
Los Angeles (LA)	118,088	4,238	5,609	27.86
Seattle (SE)	44,960	1,523	2,180	29.52
San Francisco (SF)	84,494	3,285	3,605	25.72
London (LO)	45,270	2,182	1,922	20.75
Amsterdam (AM)	49,722	1,855	1,895	26.80
Bandung (BA)	23,581	1,476	996	15.98
Jakarta (JA)	50,875	3,123	1,995	16.29
* NYC	* 225,782	* 6,442	* 7,453	* 35.05



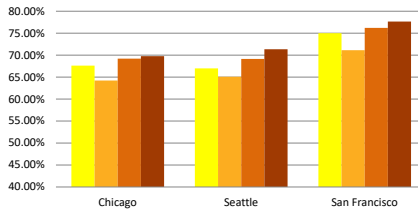
(a) Crime rate prediction accuracy



(b) Crime rate prediction f1 score



(c) Crime occurrence prediction accuracy



(d) Crime occurrence prediction f1 score

Figure 7: Generalized cross-city crime rate and occurrence prediction in U.S. cities.

Table 4: Crime Statistics in U.S. Cities

City	Training	Test	Low:(Medium):High	No:Yes
CH	4,668	964	36.7%:32.1%:31.2%	48.9%:51.1%
LA	5,488	1,206	59.8%:40.2%	NA
SE	1,588	324	33.2%:33.5%:33.3%	50.4%:49.6%
SF	2,280	476	31.7%:38.1%:30.2%	40.0%:60.0%

4.5.2 Generalization of Crime Prediction. Due to the lack of publicly available crime statistics, generalization experiments are only conducted for U.S. cities. The problem scenario and experimental settings remain unchanged as described in Section 4.4. Locally collected and processed⁴⁵⁶⁷, crime types vary among cities. Same as that in NYC, we implement 3-grade crime rate prediction with different crime rate thresholds in each city, except for Los Angeles, for which we conduct 2-taxonomy experiment since only a very limited amount of crimes are recorded. This is also the reason for the lack of a single frequent crime type in Los Angeles while the other crime data sets report locally frequent crime types. For brevity’s sake, we will focus on *Criminal Damage* for Chicago, *Vandalism* for San Francisco and *Property Damage* for Seattle as they are locally common and classify the neighborhoods more evenly than other crime types. Table 4 lists the training size, test size, crime rate ratio, and crime occurrence ratio in each city.

Figure 7 shows the prediction results. In terms of the accuracy, local embedding models perform best in all cases, followed by the NYC-based embedding models and baseline models in sequence. In particular, the NYC-based models only fall behind the local model by less than 2% for crime rate prediction and the gap is even smaller in crime occurrence prediction. As for f1-score, local embedding model still outperforms all the other methods. Random guessing in some cases results in a comparable performance to NYC-based embedding model but this is mainly due to its high recall.

The generalization experiments answer **RQ4** by demonstrating only gentle transfer errors. Like before, we conduct McNemar’s test and ascertain that the performance improvement from baseline results to the embedding model are significant at the level of 0.05. This property saves us from time-consuming local training while maintaining competitive performance with respect to all baselines.

⁴<https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2>

⁵<https://data.seattle.gov/Public-Safety/Crimes-2010/q3s4-jm2b>

⁶<http://shq.lasdnews.net/CrimeStats/CAASS/desc.html>

⁷<https://data.sfgov.org/Public-Safety/SFPD-Incidents-from-1-January-2003/tmnf-yvry>

Table 5: Location Recommendation Performance in Eight Cities.

		pre@1	pre@5	pre@10	rec@1	rec@5	rec@10	acc@1	acc@5	acc@10	MAP@1	MAP@5	MAP@10
CH	LRT	0.427	0.075	0.053	0.014	0.032	0.049	0.011	0.032	0.044	0.011	0.026	0.032
	Rank-GeoFM	0.501	0.118	0.066	0.023	0.047	0.061	0.019	0.052	0.065	0.019	0.038	0.054
	GT-SEER	0.553	0.121	0.076	0.026	0.053	0.069	0.028	0.071	0.099	0.028	0.062	0.077
	TA-PLR	0.672	0.196	0.128	0.037	0.065	0.082	0.079	0.144	0.182	0.079	0.108	0.117
	STT	0.685	0.203	0.132	0.041	0.069	0.085	0.071	0.157	0.205	0.071	0.112	0.12
	STES(NYC)	0.729	0.238	0.143	0.056	0.088	0.104	0.121	0.197	0.231	0.121	0.145	0.152
	STES(local)	0.736	0.257	0.16	0.061	0.094	0.112	0.125	0.205	0.239	0.125	0.151	0.163
LA	LRT	0.427	0.076	0.032	0.011	0.028	0.03	0.017	0.049	0.062	0.017	0.035	0.048
	Rank-GeoFM	0.464	0.09	0.051	0.014	0.033	0.036	0.023	0.076	0.107	0.023	0.049	0.057
	GT-SEER	0.498	0.115	0.076	0.021	0.039	0.042	0.036	0.095	0.131	0.036	0.062	0.076
	TA-PLR	0.524	0.137	0.096	0.023	0.045	0.051	0.05	0.121	0.149	0.05	0.067	0.084
	STT	0.523	0.149	0.102	0.024	0.048	0.056	0.053	0.123	0.154	0.053	0.071	0.082
	STES(NYC)	0.61	0.231	0.15	0.055	0.086	0.099	0.112	0.185	0.215	0.112	0.131	0.137
	STES(local)	0.617	0.232	0.154	0.059	0.096	0.111	0.114	0.189	0.221	0.114	0.139	0.142
SF	LRT	0.258	0.063	0.049	0.011	0.018	0.027	0.019	0.042	0.075	0.019	0.36	0.052
	Rank-GeoFM	0.307	0.072	0.058	0.016	0.024	0.041	0.021	0.069	0.093	0.021	0.047	0.063
	GT-SEER	0.334	0.085	0.063	0.019	0.038	0.046	0.039	0.082	0.113	0.039	0.059	0.076
	TA-PLR	0.387	0.129	0.088	0.025	0.046	0.058	0.054	0.112	0.159	0.054	0.079	0.093
	STT	0.396	0.141	0.095	0.032	0.051	0.06	0.062	0.121	0.158	0.062	0.084	0.091
	STES(NYC)	0.425	0.172	0.122	0.048	0.070	0.082	0.098	0.154	0.182	0.098	0.112	0.117
	STES(local)	0.432	0.176	0.125	0.054	0.075	0.089	0.103	0.162	0.186	0.103	0.117	0.124
SE	LRT	0.452	0.088	0.057	0.013	0.019	0.048	0.01	0.063	0.079	0.01	0.024	0.046
	Rank-GeoFM	0.478	0.111	0.065	0.018	0.026	0.057	0.028	0.081	0.109	0.028	0.039	0.062
	GT-SEER	0.512	0.143	0.079	0.028	0.037	0.066	0.037	0.101	0.145	0.037	0.056	0.074
	TA-PLR	0.566	0.189	0.105	0.032	0.043	0.075	0.05	0.131	0.189	0.05	0.074	0.091
	STT	0.574	0.205	0.112	0.037	0.05	0.081	0.069	0.153	0.196	0.069	0.086	0.097
	STES(NYC)	0.643	0.252	0.167	0.058	0.091	0.104	0.121	0.205	0.242	0.121	0.143	0.15
	STES(local)	0.645	0.257	0.169	0.054	0.091	0.106	0.124	0.215	0.252	0.124	0.147	0.156
LO	LRT	0.324	0.052	0.035	0.017	0.029	0.034	0.018	0.049	0.083	0.018	0.026	0.037
	Rank-GeoFM	0.385	0.07	0.053	0.024	0.036	0.041	0.027	0.076	0.112	0.027	0.048	0.062
	GT-SEER	0.397	0.089	0.054	0.027	0.042	0.049	0.041	0.094	0.131	0.041	0.065	0.078
	TA-PLR	0.428	0.124	0.069	0.038	0.056	0.061	0.073	0.158	0.182	0.073	0.101	0.112
	STT	0.432	0.127	0.075	0.042	0.055	0.064	0.079	0.158	0.189	0.079	0.098	0.109
	STES(NYC)	0.501	0.159	0.097	0.051	0.07	0.079	0.131	0.193	0.221	0.131	0.147	0.153
	STES(local)	0.514	0.169	0.103	0.062	0.081	0.094	0.141	0.216	0.245	0.141	0.158	0.163
AM	LRT	0.523	0.074	0.053	0.016	0.032	0.043	0.029	0.101	0.13	0.029	0.047	0.062
	Rank-GeoFM	0.605	0.119	0.068	0.023	0.047	0.056	0.042	0.123	0.153	0.042	0.074	0.08
	GT-SEER	0.632	0.136	0.085	0.036	0.063	0.079	0.061	0.143	0.171	0.061	0.097	0.105
	TA-PLR	0.715	0.208	0.122	0.048	0.093	0.116	0.082	0.217	0.273	0.082	0.146	0.159
	STT	0.702	0.195	0.116	0.048	0.089	0.123	0.101	0.205	0.252	0.101	0.154	0.163
	STES(NYC)	0.766	0.234	0.134	0.077	0.112	0.128	0.178	0.267	0.304	0.178	0.2	0.207
	STES(local)	0.776	0.253	0.139	0.087	0.126	0.141	0.189	0.283	0.322	0.189	0.213	0.225
JA	LRT	0.289	0.065	0.048	0.012	0.021	0.027	0.01	0.033	0.052	0.01	0.016	0.029
	Rank-GeoFM	0.325	0.067	0.052	0.017	0.026	0.033	0.013	0.040	0.069	0.013	0.026	0.038
	GT-SEER	0.421	0.086	0.059	0.022	0.034	0.041	0.023	0.062	0.087	0.023	0.035	0.058
	TA-PLR	0.488	0.099	0.053	0.027	0.049	0.05	0.032	0.082	0.154	0.032	0.047	0.065
	STT	0.532	0.116	0.063	0.028	0.057	0.068	0.048	0.113	0.179	0.048	0.065	0.084
	STES(NYC)	0.581	0.162	0.091	0.059	0.082	0.093	0.124	0.173	0.19	0.124	0.137	0.141
	STES(local)	0.596	0.176	0.108	0.062	0.097	0.106	0.135	0.189	0.218	0.135	0.151	0.165
BA	LRT	0.405	0.053	0.037	0.015	0.028	0.034	0.012	0.028	0.046	0.012	0.02	0.028
	Rank-GeoFM	0.452	0.059	0.044	0.018	0.036	0.047	0.018	0.043	0.061	0.018	0.026	0.037
	GT-SEER	0.493	0.082	0.05	0.021	0.046	0.054	0.024	0.062	0.083	0.024	0.031	0.056
	TA-PLR	0.555	0.12	0.078	0.036	0.053	0.062	0.044	0.118	0.168	0.044	0.06	0.071
	STT	0.583	0.165	0.102	0.052	0.067	0.089	0.061	0.134	0.182	0.061	0.082	0.097
	STES(NYC)	0.641	0.215	0.144	0.069	0.092	0.107	0.121	0.179	0.193	0.121	0.148	0.153
	STES(local)	0.662	0.234	0.167	0.079	0.108	0.127	0.138	0.207	0.223	0.138	0.161	0.17

5 CONCLUSIONS

In this paper, we propose an unsupervised embedding model that learns to represent social network check-ins and their functional, temporal, and geographic aspects in the form of dense numerical vectors in a semantic space. Item correlations are well captured in terms of activity and location similarities.

We show two model applications *location recommendation* and *crime prediction*. Our embedding model based recommendation algorithm STES outperforms a wide range of state-of-the-art methods according to four performance metrics. Crime prediction demonstrates the model's effectiveness at capturing location properties

and verifies the possibility to infer crime rates or occurrences from residents' daily activities. Furthermore, we confirm that the embedding model has good generality and can be trained in well-represented cities before being applied in other places with only a small generalization error.

There are several interesting lines of future investigation: (1). On the theoretical side, we would like to explore whether the embedding model can be fit into a supervised deep learning framework (e.g., a convolutional or recurrent neural network) for further performance improvement. We may also include some other messages during the embedding training process, e.g., users' demographic

information and social relations, to describe places and users from more perspectives. (2). On the application side, we would like to deploy the model to more tasks such as urban functional zone study and travel time inference.

ACKNOWLEDGMENTS

We thank Zhiyuan Cheng for providing us with the check-in data set and Bo Hu for sharing their location recommendation algorithm details.

REFERENCES

- [1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
- [2] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks.. In *Aaai*, Vol. 12. 17–23.
- [3] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where You Like to Go Next: Successive Point-of-Interest Recommendation.. In *IJCAI*, Vol. 13. 2605–2611.
- [4] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z Sui. 2011. Exploring Millions of Footprints in Location Sharing Services. *ICWSM* 2011 (2011), 81–88.
- [5] Andrew Church, Martin Frost, and Keith Sullivan. 2000. Transport and social exclusion in London. *Transport Policy* 7, 3 (2000), 195–205.
- [6] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM conference on Recommender systems*. ACM, 93–100.
- [7] Matthew S Gerber. 2014. Predicting crime using Twitter and kernel density estimation. *Decision Support Systems* 61 (2014), 115–125.
- [8] Tiek He, Hongzhi Yin, Zhenyu Chen, Xiaofang Zhou, Shazia Sadiq, and Bin Luo. 2016. A Spatial-Temporal Topic Model for the Semantic Annotation of POIs in LBSNs. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 1 (2016), 12.
- [9] Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J Smola, and Kostas Tsoutsoulouklis. 2012. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 769–778.
- [10] Bo Hu, Mohsen Jamali, and Martin Ester. 2013. Spatio-temporal topic modeling in mobile social media for location recommendation. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 1073–1078.
- [11] Rizwan Iqbal, Masrah Azrifah Azmi Murad, Aida Mustapha, Payam Hasanyan Shariat Panahy, and Nasim Khanahmadiravi. 2013. An experimental study of classification algorithms for crime prediction. *Indian Journal of Science and Technology* 6, 3 (2013), 4219–4225.
- [12] Donald E Knuth. 1985. Dynamic huffman coding. *Journal of algorithms* 6, 2 (1985), 163–180.
- [13] Yehuda Koren, Robert Bell, Chris Volinsky, and others. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [14] John Krumm and Dany Rouhana. 2013. Placer: semantic place labels from diary data. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, 163–172.
- [15] Takeshi Kurashima, Tomoharu Iwata, Takahide Hoshida, Noriko Takaya, and Ko Fujimura. 2013. Geo topic model: joint modeling of user’s activity area and interests for location recommendation. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 375–384.
- [16] Jia Li, Hua Xu, Xingwei He, Junhui Deng, and Xiaomin Sun. 2016. Tweet modeling with LSTM recurrent neural networks for hashtag recommendation. In *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 1570–1577.
- [17] Xutao Li, Gao Cong, Xiao-Li Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-GeoFM: a ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 433–442.
- [18] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 831–840.
- [19] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. 2013. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1043–1051.
- [20] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- [21] Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the Context of Locations for Personalized Location Recommendations. 2016 (2016), 2016.
- [22] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, 17–24.
- [23] Wesley Mathew, Ruben Raposo, and Bruno Martins. 2012. Predicting future locations with hidden Markov models. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 911–918.
- [24] Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* 12, 2 (1947), 153–157.
- [25] Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168* (2013).
- [26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*. 3111–3119.
- [27] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. 2011. Exploiting Semantic Annotations for Clustering Geographic Areas and Users in Location-based Social Networks. *The social mobile web* 11 (2011), 02.
- [28] Makbule Gulcin Ozsoy. 2016. From Word Embeddings to Item Recommendation. *arXiv preprint arXiv:1601.01356* (2016).
- [29] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 701–710.
- [30] Van-Thuy Phi, Liu Chen, and Yu Hirate. 2016. Distributed Representation-based Recommender Systems in E-commerce. 2016 (2016), 2016.
- [31] Ruslan Salakhutdinov and Andriy Mnih. 2011. Probabilistic matrix factorization. In *NIPS*, Vol. 20. 1–8.
- [32] Siddharth Sarda, Carsten Eickhoff, and Thomas Hofmann. 2016. Semantic Place Descriptors for Classification and Map Discovery. *arXiv preprint arXiv:1601.05952* (2016).
- [33] Salvatore Scellato, Anastasios Noulas, and Cecilia Mascolo. 2011. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1046–1054.
- [34] Yang Song, Ali Mamdouh Elkahky, and Xiaodong He. 2016. Multi-rate deep learning for temporal recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 909–912.
- [35] Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proc. ACL*.
- [36] Duyu Tang, Bing Qin, Ting Liu, and Yuekui Yang. 2015. User modeling with neural network for review rating prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 1340–1346.
- [37] Mozghan Tavakolifard and Kevin C Almeroth. 2012. Social computing: an intersection of recommender systems, trust/reputation systems, and social networks. *IEEE Network* 26, 4 (2012), 53–58.
- [38] Peter J Van Koppen and Robert WJ Jansen. 1999. The time to rob: variations in time of number of commercial robberies. *Journal of Research in Crime and Delinquency* 36, 1 (1999), 7–29.
- [39] Svitlana Volkova and Eric Bell. 2016. Account Deletion Prediction on RuNet: A Case Study of Suspicious Twitter Accounts Active During the Russian-Ukrainian Crisis. 2016 (2016), 2016.
- [40] Xiaofeng Wang, Matthew S Gerber, and Donald E Brown. 2012. Automatic crime prediction using events extracted from twitter posts. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer, 231–238.
- [41] David Weisburd, Shawn Bushway, Cynthia Lum, and Sue-Ming Yang. 2004. Trajectories of crime at places: A longitudinal study of street segments in the city of Seattle. *Criminology* 42, 2 (2004), 283–322.
- [42] Sanjaya Wijeratne, Lakshika Balasuriya, Derek Doran, and Amit Sheth. 2016. Word embeddings to enhance twitter gang member profile identification. In *IJCAI Workshop on Semantic Machine Learning (SML 2016)*. New York City, NY: CEUR-WS, Vol. 7. 2016.
- [43] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin, and Krzysztof Janowicz. 2011. On the semantic annotation of places in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 520–528.
- [44] Mao Ye, Peifeng Yin, and Wang-Chien Lee. 2010. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. ACM, 458–461.
- [45] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. 2014. LORE: exploiting sequential influence for location recommendations. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 103–112.
- [46] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R Lyu. 2016. GT-SEER: Geo-Temporal SEquential Embedding Rank for Point-of-interest Recommendation. *arXiv preprint arXiv:1606.05859* (2016).
- [47] Zack Zhu, Jing Yang, Chen Zhong, Julia Seiter, and Gerhard Tröster. 2015. Learning functional compositions of urban spaces with crowd-augmented travel survey

data. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 22.